Discuss the following topics on Java Database Programming. Select at least **one** of the following topics for your initial post.

* Batch Processing.
* Scrollable and Undated ResultSet.
* Scrollable and Undated RowSet.
* Storing and Retrieving Images in JDBC.

As we have previously learned, JDBC, also known as Java Database Connectivity, is an API used to communicate or connect with a database while retrieving data from the database or even updating it. It does this through SQL queries and uses interfaces to help make the progress possible. It is a very powerful tool and acts as a bridge between different programming languages. Part of the capabilities of JDBC allows for a scrollable and updated ResultSet.

The ResultSet is a vital part of database connectivity (GeeksforGeeks, 2024). ResultSet is an object that "represents the result of a SQL Query executed on a database" (GeeksforGeeks, 2024). It helps to navigate through the resulting data and update it as needed (GeeksforGeeks, 2024). The ResultSet can be thought of as a spreadsheet or table with rows representing records and columns representing database fields (GeeksforGeeks, 2024). A ResultSet goes through each row and can be brought to a specific row using a method such as next() (GeeksforGeeks, 2024). Data can be updated using ResultSet through different methods such as "updateInt(int columnIndex, int x), updateString(int columnIndex, String x), updateBoolean(int columnIndex, boolean x), updateRow(), and deleteRow()" (GeeksforGeeks, 2024).

There are three types of ResultSet: scrollability, concurrency, and holdability (GeeksforGeeks, 2024). Scrollability is what allows the cursor in ResultSet to move around as commanded (GeeksforGeeks, 2024). It can move forward through rows by using TYPE\_FORWARD\_ONLY, scroll forward and backward, but not "reflect ResultSet" using TYPE\_SCROLL\_INSENSITIVE, and move forward and backward, but changes the ResultSet using TYPE\_SCROLL\_SENSITIVE (GeeksforGeeks, 2024).

**Reference**

GeeksforGeeks. (2024, May 27). *JDBC Result Set*. GeeksforGeeks. https://www.geeksforgeeks.org/jdbc-result-set/

**Assignment Requirements and Grading:**

* 1. An initial post of approximately 250 words is due by **Thursday, 11:59 p.m., CST**.
  2. For the initial post to be considered substantive, it should be at least 250 words in length and fully cover the topics being presented. Single-sentence definitions or responses will not be awarded points.
  3. Submit your post by clicking on the assignment link above, then Create Thread. You must create a thread in order to view your peers' posts. Tip: Create your post in a Word document and then copy and paste your work into the thread.
  4. A minimum of three (3) responses, to the original threads of other students, of 100-200 words each are due by **Sunday, 11:59 p.m., CST**.
  5. To view the rubric grading criteria, click on the following link: [Discussion Board Grading Rubric.](https://content.bellevue.edu/cst/csd/rubricdbv3.pdf)

Hey, Megan. I think you did a nice job on your discussion board. You did a nice job explaining how storing and retrieving images in JDBC works. I am glad you mentioned the different types of large objects because understanding the difference helps decide which is best. The sample code you included does a good job of expanding on your initial thoughts. GeeksforGeeks often helps me understand concepts that do not always click the first time reading through them. I can see how using Large Objects in our code can be helpful. If we already know that an image, video, or whatever data will be larger, we can plan for that using a large object.

Hello there, Brian! Your discussion board for this week is very thorough, and you do an excellent job of explaining batch processing in Java. I really like how you started by describing how you think batch processing is powerful yet underrated. I agree that batch processing can be beneficial for developing background running tasks. JobRunr seems like a handy tool in Java, and I could see myself utilizing it in the future since I have not experimented with it yet. I completely agree that this seems like a significant time saver. Your included code does a nice job of showing how it works.

Hey, Colton! I enjoyed reading your post for this week's module; I think you did a great job explaining ResultSet and scrollable in JDBC. I also decided to write about the same topic, so I was intrigued to read your explanation since it always seems to provide further insight. I completely agree with everything that you mention in your post. Your inclusion of those segments of code does a great job of demonstrating the concepts you explained. I like how even if a capability itself in developing does not seem versatile, it can usually be used with it to extend the range of abilities.